

The challenges of SVM optimization using Adaboost on a phoneme recognition problem

Rimah Amami*, Dorra Ben Ayed[†] and Nouredine Ellouze[‡]

Department of Electrical Engineering

National School of Engineering of Tunis, University of Tunis - El Manar, Tunisia

*Email: Rimah.amami@yahoo.fr

Abstract—The use of digital technology is growing at a very fast pace which led to the emergence of systems based on the cognitive infocommunications. The expansion of this sector impose the use of combining methods in order to ensure the robustness in cognitive systems.

Boosting is a technique for combining many weak classifiers to form one high-performance prediction rule to improve the performance of any given learning algorithm. In theory, Boosting can be used to reduce the error of the learning algorithm which generates learners that needs to be better than random guessing. Adaboost is the most used boosting algorithm. Support Vector Machines (SVMs) and Adaptive Boosting (Adaboost) are two successful classification methods which are essentially similar as they both try to maximize the minimal margin on a training set. In this paper, we will investigate the impact of Adaboost on the supervised algorithm SVM for a Multi-Class phoneme recognition problem. This task may be complex since SVM is not an easy classifier to train and so, AdaboostSVM may not be viable. Furthermore, we compare the recognition rates to other commonly used Adaboost methods, such as Decision Tree C4.5. From the different phoneme datasets, we shall show that the single SVM-RBF outperforms Decision Tree C4.5, the AdaboostC4.5 and even the Adaboost SVM-Based component classifier.

To experiment AdaboostSVM, we use the phoneme datasets from TIMIT corpus and MFCC feature representations.

Index Terms—SVM, Adaboost, Optimization, C4.5, phoneme

I. INTRODUCTION

Speech recognition proved to be successful in enormous applications in the last few decades. The Cognitive Infocommunications (CogInfoCom) systems ties in closely with the patterns and speech recognition area since the user and the CogInfoCom systems can interact through a human computer communication [1] [2].

There has been a considerable improvement in speech applications since it is closely related to the information and communications technology and cognitive science problems.

Despite the considerable progress in speech researches, the robustness of speech recognition systems still slight due to many factors such as large data, noisy environments, speaker voice, etc. Unfortunately, many speech recognition systems have their limits to accomplish the human performance.

For example, a phoneme recognition system is based on a sequence of phones but those phones bring the understanding and the confusion with the neighboring phonemes. Thus, many studies in the speech recognition develop techniques that provide into the speech recognition systems greater

levels of knowledge of the language. Indeed, the relationship between speech recognition and cognitive capacity proved to have a direct impact on the human-interaction systems. But how is the interaction between speech and cognitive science in the development of speech recognition systems?

In fact, the cognitive model based on the speech recognition systems have to better focusing in the descriptions input such as the raw acoustic signal in order to boost the accuracy of the recognition system. In this context, this paper tries to improve the performance of a speech recognition system based on a raw acoustic input recording from a microphone. Several speakers (female and male) pronounce few sentences but each speaker has its own behavior, accent, energy, noise, etc. Those cognitive factors make it difficult to the robustness of the speech recognition system.

In the automatic speech recognition (ASR) field, the choice of the learning algorithm for building of any ASR system is a crucial step since the success of the recognition task depends essentially on learning stage. Recently, the Boosting have been proved to be an efficient method for improving the performance of different classifiers. Therefore, this technique is used to combine a collection of component classifiers (called also weak classifiers), to form a single "strong" classifier. It consists in calling repeatedly the component classifier on different weights over the training samples and adaptively adjusting these weights after each Boosting iteration [8].

Adaboost is the most popular method of Boosting. The great success of Adaboost can be attributed to its ability to maximize the margin on a training set, which lead to improve the performance of the classifier. The efficiency of Adaboost is closely related to the component classifier used. Since the main objective of Adaboost is to enhance the learning performance of a given weak classifier, then combining Adaboost with a strong classifier may not necessarily be optimum choice and it may be going against the gain of the Boosting principle. Furthermore, the choice of the learning algorithm, generally, affect the recognition rates. But how to distinguish between a "weak" and a "strong" learner algorithm ?

We propose this study to emphasize the notion of "the weak and strong learner"; SVM and C4.5 are two strong algorithms

which constitutes our base component classifiers. In particular, we would observe the behaviour of SVM with Adaboost.

The remaining parts of this paper is organized as follows: In section II, the main idea of Adaboost is introduced. In section III, the learning algorithms used in this study are presented; The architecture of our ASR system is proposed in section IV; Experimental setup and results are described in sections V. The conclusion is made in section VII.

II. ADABOOST

The Boosting method was inspired by on-line learning algorithm called Hedge (β) which allocate weights to a set of hypothesis used to predict the outcome of any system [8]. Recently, Boosting method has been quite successfully applied on real-world applications and it was mostly used on the speech recognition field. The most and widely used boosting algorithm is AdaBoost.

AdaBoost (Adaptive Boosting), an iterative algorithm, was originally introduced by Freund and Schapire on 1995 [8] [9]. If the applied learning algorithm had a low performance, Adaboost algorithm generates a sequentially weighted set of weak classifiers in order to create new classifiers which are more operational on the training data. Hence, the AdaBoost algorithm multiple iteratively classifiers to improve the classification accuracies of many different data sets compared to the given best individual classifier.

The main idea of Adaboost is to run repeatedly a given weak learning algorithm in different probability distributions, W , over the training data. This distribution is initially set uniform.

In the meantime, Adaboost calls the Weak Learner algorithm repeatedly in a series of cycles T . Then, it assigns higher weights to the misclassified samples by the current component classifier (at cycle t), in the hopes that the new weak classifier can reduce the classification error by focusing on it. Meanwhile, lower weights will be assigned to the correctly classified samples [7] [10]. Thereafter, the distribution W is updated after each cycle. In the end, hypothesis produced by the weak learner from each cycle are combined into a single "Strong" hypothesis f [16].

The importance of Adaboost lies in the component classifiers which is systematically have an accuracy slightly higher than 50%. This means that the component classifiers have to be better than a random estimation [15].

Briefly, Adaboost is a meta-learning method that tries to build a "strong" learning algorithm based on a group of "weak" classifiers. It must be pointed out that Boosting has been very successful for solving two-class classification problems.

Meanwhile, to achieve multi-class classification, most algorithms have to convert the multi-class problem to a multiple binary classification problems. Our ASR system will use Adaboost.M1.

Since this paper discuss an hybrid of the learning algorithm with Adaboost, we will present an overview of other related studies that used Adaboost in cognitive systems. In the few past years, several studies combining Adaboost and cognitive systems have been developed to take advantage of the Adaboost algorithm to improve the system's performance. Chakraborty present in [17] an expert cognitive system which use Adaboost to boosts the performance of an ensemble of classifiers. The empirical comparison of his study shows that hybrid learner based Adaboost outperforms the single weak learner.

Stanculescu et al. used Adaboost algorithm in order to improve the real-time object detection in complex robotics problem [18]. Experimental results of this study on a car database show that the boosted classifier improve the results of the vehicle-detection application.

On the other hand, Lee and al. used Adaboost for the text detection in natural scene to enhance the detection system [19].

For a driver's cognitive distraction detection problem, the authors used Adaboost to improve the performance for detection of driver distraction [20]. Based on experimental results, the authors shows the capability of AdaBoost to enhance the accuracy of a problem based on the detection of state of cognitive distraction.

Adaboost was combined with SVM for a triaxial accelerometer-based fall detection problem in [21]. The experimental results proves that the proposed method Adaboost-SVM gives optimal results compared to those with the single method. The Adaboost-SVM produces, also, the the lowest false alarm rate and the detection results as well as the highest accuracy results of their study.

III. COMPENENT CLASSIFIERS

The principal aim of the learning algorithm is to extract regularities from sets of samples. They are consisting of several algorithms that improve automatically the system through experience [27].

In this paper, we are interested by the supervised learning since it allows to generate function that maps inputs to desired outputs.

In the last decades, different learning algorithms were applied to identify and verify the speech. There is several algorithms that were mostly used in the ASR such as decision tree, LDA, SVM, the baseline system, etc.

A. Support Vector Machine

A Support Vector Machine (SVM) is a learning algorithm for pattern recognition and regression problems [25] whose approaches the classification problem as an approximate implementation of the Structural Risk Minimization (SRM) induction principle [4].

SVM approximates the solution to the minimization problem

of SRM through a Quadratic Programming optimization. It aims to maximize the margin which is the distance from a separating hyperplane to the closest positive or negative sample between classes [3].

A subset of training samples is chosen as support vectors. They determine the decision boundary hyperplane of the classifier. Based on this principle, the SVM adopts a systematic approach to find a linear function that belongs to a set of functions with lowest VC dimension (the VapnikChervonenkis dimension measure the capacity of a statistical classification algorithm).

Applying a kernel trick that maps an input vector into a higher dimensional feature sapce, allows to SVM to approximate a non-linear function [4] and [15] and [3]. In this paper, we use SVM with the radial basis function kernel (RBF).

B. Decision Tree C4.5

C4.5 is a learning algorithm used to generate a decision tree developed by Ross Quinlan [23]. The Decision Tree C4.5 is an extension of Quinlan's earlier ID3 algorithm (Iterative Dichotomiser 3).

Just like ID3, C4.5 employs a "divide and conquer" strategy [13] and uses the concept of information entropy to compute builds decision trees from a set of training (the split criteria) The training data is a set $X = x_1, x_2, \dots, x_n$ of already classified samples. Each sample $X_i = x_1, x_2, \dots$ is a vector where x_1, x_2, \dots represent attributes or feature of the sample.

The training data is augmented with a vector $C = c_1, c_2, \dots$ where c_1, c_2, \dots represent the class to which each sample belongs [22].

At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision.

With the adaptively boosted C4.5 Decision Tree classifiers [27], very high degree of accuracy can be achieved in ASP field.

IV. PHONEME RECOGNITION SYSTEM ARCHITECTURE

The architecture of our phoneme recognition system is described in this section. The proposed system aims to compare the performance of Adaboost with SVM-based component classifier and the performance of Adaboost with Decision Tree C4.5-based component classifier on a phoneme recognition task,(see Fig. 1).

A. Adaboost-Component Classifier

The proposed phoneme recognition system passes through several stage with the transformation of the speech samples

to spectrogram and then to MFCC spectrum by applying the Spectral analysis. Then, we proceed to the the phoneme segmentation to constitute seven sub-phoneme sets. The finale stage consists on applying the proposed algorithms for the phoneme recognition problem, see figure 1.

This system aims to use two different algorithms as component classifier in Adaboost. Then, the comparison of error generalization is made in order to see the ability of Adaboost with those two classifiers.

V. EXPERIMENTAL SETUP

The experiments with our ASR system consists in recognizing which identity of phoneme which been tested (i.e. aa, ae, ih , etc).

We split beforehand the phonemes into 7 groups (Vowels,Stops,Nasals,Fricatives,Affricates, Semi-vowels, others(silence)) [3].

As discussed in the previous section, the first setp in a ASR system is the feature extraction. It converts the speech waveform to a set of parametric representation. Hence, we have used the MEL frequency cepstral coefficients (MFCC) feature extractor.

In 1980, Davis and Mermelstein developed the MFCC features for speech classification systems [6]. It consists on the cepstral coefficients which are produced by the mel-frequency warped Fourier transform function.

The system that we present in this paper use the speech samples extracted from the TIMIT corpus [11]. Moreover, for the nonlinear SVM approach with the one-against-one strategy , we choose the RBF (Gaussian) Kernel trick, this choice was made after a previous study done on our data sets with different kernel tricks (Linear, Polynomial, Sigmoid) [3].

The experiments using SVM are done using LibSVM toolbox [29]. The table I recapitulate our main choice of experiments conditions:

TABLE I
EXPERIMENTAL SETUP

Methods	-SVM -C4.5 -AdaboostSVM -AdaboostC4.5
γ	1/39
Cost	10
Kernel trick	RBF
Windowing	3-middle Windows
Corpus	TIMIT
Dialect	New England
Frame rate	125/s
Features number	39
Sampling frequency	16ms

VI. EXPERIMENTAL RESULTS

In this section, SVM and Decision Tree C4.5 are compared with the commonly used Adaboost, which takes SVM and

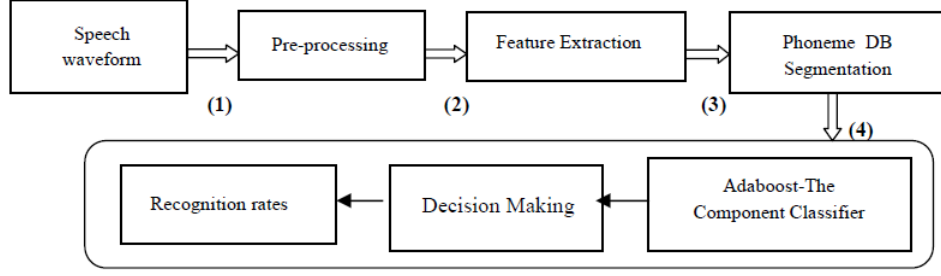


Fig. 1. Architecture of the ASR system

Decision Tree C4.5 as component classifiers.

At this stage of experiments, we investigate to find the impact of Adaboost on the two learning classifiers selected.

For the phoneme recognition systems, we compare the performance of our component classifier in Adaboost and single classifiers on 7 data sets which are vowels, semi-vowels, stops, others, nasals, fricatives, affricates. The final performance of each algorithm on a data set is the average of the results over the 7 data sets.

It must be pointed out that the number of iteration of Adaboost for the recognition system was fixed to 25. In general, the boosted classifier performs well even with only 10 iterations.

TABLE II
GENERALIZATION ERRORS WITH 3-MIDDLE FRAMES OF FOUR ALGORITHMS PER PHONEME : SVM-RBF, ADABOOST WITH SVM, DECISION TREE C4.5 AND ADABOOST WITH DECISION TREE C4.5

Classifiers	SVM	AdaboostSVM	C4.5	AdaboostC4.5
Vowel	44.74	45.83	72.95	75.80
Semi-Vowel	18.55	22.38	38.91	27.62
Stops	45.72	45.99	64.62	68.45
Others	14.93	15.97	18.40	16.32
Nasal	39.46	41.57	61.45	48.80
Fricative	21.02	26.14	44.51	29.17
Affricate	21.43	33.33	45.24	33.33
Average	29.40	32.95	49.44	42.78

The table II describes generalization errors of the four algorithms for the given 7 data sets with 3-middle frames which are Vowel, Semi-Vowel, Fricative, Affricate, Nasal, Silences and Stops .

The empirical results present the performance of the phoneme recognition system with the singles classifiers SVM and C4.5 and the combined classifiers. The database is composed by 70% of learning data and 30% of test data

which are used for the validation.

Since the generalization errors of AdaboostC4.5 are less than those of C4.5 on these 7 data sets. We can conclude that AdaboostC4.5 performs better than C4.5 (i.e. Nasal : 61% Vs 48%).

In Semi-Vowel, Others, Nasal, Fricative, Affricate, AdaboostC4.5 gets better accuracy compared with those generates by C4.5 (i.e. the accuracy of Fricative data sets improves about 15%).

In turn, the same conclusion can not be drawn from table II for the single SVM-RBF and AdaboostSVM. We observe that the AdaboostSVM performs similarly or worse than the single SVM-RBF. We think that forcing the strong SVM classifiers (SVM with its best parameters) to concentrate on the very hard samples with too much emphasis is the cause of performance degradation in AdaboostSVM algorithm. This case is also observed in [32] which show that Adaboost with strong classifier component classifier is not effective.

For AdaboostSVM, the accuracy have been slightly better on vowel, stops and affricate data sets and the improvement reach about 1,8%. While, for the rest of data sets used, AdaboostSVM performs slightly worse than single SVM and the declination reach about 4%.

From table II, we, also, observed that SVM outperforms both C4.5 and Adaboost C4.5, in general, on these data sets. Furthermore, for SVM we set a small value of γ and also the most suitable for our data ($\gamma=0.008$), which make the SVM classifier stronger.

Thus, Adaboost become inefficient because the errors of these component classifiers are highly correlated. Hence, the use of a suitable gamma for SVM component classifier un Adaboost leads to lower the performance of AdaboostSVM for phoneme recognition.

We would like to emphasize that the purpose of our experiments is not to argue that SVM-RBF performs better than the decision tree C4.5 and the boosted C4.5, but rather to illustrate that SVM and C4.5 are two strong algorithm, but

with SVM optimization problem has been encountered since the performance degradation is the natural expectation of the hybrid Adaboost and the strong SVM-RBF to the multi-class case.

In the last decades, some experiments were carried out in order to better understand why boosting sometimes leads to a deterioration in generalization performance. Freund and Schapire put this down to overfitting a large number of trials T (T for Adaboost iteration number) which allows to composite classifier to become very complex [9] and [24].

Besides these, boosting try to build a strong classifier from weak classifiers. However, if the performance of the classifier is better than random estimation, then boosting may not result in a strong classifier and this method would be going against the gain of the Boosting principle and not achieve the desired results [33].

VII. CONCLUSION

In this paper, we have used Adaboost as classifier in order to build a phoneme recognition system. We combined Adaboost algorithm with both SVM and C4.5 classifiers in order to show the impact of this algorithm on the performance of recognition system. For the this phonemes recognition system we have shown that AdaboostC4.5 have on error rate lower than the single C4.5. But, single SVM-RBF outperforms AdaboostSVM.

Boosting seems to be effective when applied to Decision Tee C4.5 than SVM-RBF although the performance of the adaboosted C4.5- based component classifier is less than single SVM.

Hence, boosting can not improve the performance of SVMs, and we guess the reason of this phenomenon is that SVMs is a strong classifier. We also found Adaboost with SVM always achieves worse results on all of the phoneme collections. The future work will be to extend our work in order to improves the phoneme recognition accuracy.

REFERENCES

- [1] P. Baranyi and A. Csapo. Definition and Synergies of Cognitive Infocommunications. Acta Polytechnica Hungarica, vol. 9, pp. 67-83, 2012.
- [2] G. Sallai. The Cradle of Cognitive Infocommunications. Acta Polytechnica Hungarica, vol. 9, no. 1, pp. 171181, 2012.
- [3] R. Amami, D. Ben Ayed, and N. Ellouze. Phoneme recognition using support vector machine and different features representations. In *The 9th International Conference Distributed Computing and Artificial Intelligence (DCAI), Spain*, 2012. \$http://rd.springer.com/chapter/10.1007/978-3-642-28765-7_71\$.
- [4] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, 1995.
- [5] T. Cover. The best two independent measurements are not the two best. *IEEE Transactions On Systems Man And Cybernetics*, SMC-4(1):116–117, 1974.
- [6] S. B Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoust Speech Signal Processing*, 28(4):357–366, 1980.
- [7] Donghong Ji. Application of Boosting to Chinese Word Sense Disambiguation. *International Conference on Natural Language Processing and Knowledge Engineering*, 2005
- [8] Y. Freund and R.E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. In *Proceeding of the Second European Conference on Computational Learning Theory*, pages 23–37, 1995.
- [9] Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese society for artificial intelligence*. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1997.
- [10] Haifeng Fan, , and Haixing Wang. "Predicting protein subcellular location by AdaBoost.M1 algorithm. *2nd International Conference on Artificial Intelligence Management Science and Electronic Commerce (AIMSEC)*. 2011.
- [11] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue. Timit acoustic-phonetic continuous speech corpus. In *Texas Instruments (TI) and Massachusetts Institute of Technology (MIT)*, 1993.
- [12] R. Jang. *Audio Signal Processing and Recognition, year 2009, publisher Tsing Hua University*.
- [13] R. Kohavi and J. R. Quinlan. Decision tree discovery. handbook of data mining and knowledge discovery. *Oxford University*, pages 267–276, 2002.
- [14] F. Li, J. Koeck, and H. Wechsler. Strangeness based feature selection for part based recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop.*, page 22, 2006.
- [15] X. Li, L. Wang, and E. Sung. Adaboost with svm-based compnent classifiers. *Engineering Applications of Artificial Intelligence*, 21:785–795, 2008.
- [16] R. Meir and G. Ratsch. An introduction to boosting and leveraging. *Advanced lectures on Machine Learning, LNCS, Springer*, (119-184), 2003.
- [17] D. Chakraborty. An Expert Cognitive System Using Ada-Boost Algorithm. International Conference on Computational Science. 2005.
- [18] B. Stanculescu, A. Breheret, F. Moutarde. Introducing New AdaBoost Features for Real-Time Vehicle Detection. Conference on COGNITIVE systems with Interactive Sensors, Stanford, Palo Alto, United States. 2007.
- [19] J.J. Lee, P.H. Lee, S.W. Lee, A. Yuille, C. Koch. AdaBoost for Text Detection in Natural Scene. IEEE International Conference on Document Analysis and Recognition. pp. 429-434. 2011.
- [20] M. Miyaji, M. Danno, H. Kawanaka, K. Oguri. Driver's Cognitive Distraction Detection using AdaBoost on Pattern Recognition Basis. IEEE International Conference on Vehicular Electronics and Safety. Columbus, OH, USA. pp 22-24. 2008.
- [21] W. C. Cheng, D. M. Jhan. Triaxial Accelerometer-Based Fall Detection Method Using a Self-Constructing Cascade-AdaBoost-SVM Classifier. IEEE Journal of biomedical and health informatics. Vol. 17, No. 2. 2013.
- [22] D. Pomorski and P.B P. B Perche. Inductive learning of decision trees: application to fault isolation of an induction motor. *Engineering Applications of Artificial Intelligence*, 14(2):155–166, 2001.
- [23] J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.
- [24] J. R. Quinlan. Bagging, boosting, and c4.5. In *roceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 725–730, 1996.
- [25] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
- [26] A. Torralba, K. Murphy, and W. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–769, 2004.
- [27] V. Vapnik. The nature of statistical learning theory. *Springer-Verlag New York*, 8(6):188, 1995.
- [28] A. Vezhnevets and V. Vezhnevets. Modest adaboost teaching adaboost to generalize better. *Proc. Graphicon*, pages 322–325, 2005.
- [29] C.C. Chang, C.J. Lin. LIBSVM: a Library for Support Vector Machines, Department of Computer Science National Taiwan University, Taipei, Taiwan, 2011.
- [30] P. Viola and M. Jones. Robust real-time face detection. In *Proceedings Eighth International Journal on Computer Vision*, volume 2, pages 137–154, 2001.
- [31] Feng Zhang. A Novel Approach to Keyword Extraction for Contextual Advertising. *First Asian Conference on Intelligent Information and Database Systems*. 2009.
- [32] J. Wickramaratna, S. B. Holden, and B. F. Buxton. Performance degradation in boosting. In *Proceedings of the Second International Workshop on Multiple Classifier Systems*, page 1121, 2001.

- [33] Li, X.. AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*. 2008.

/aa/ /ae/ /ah/ /ao/ /aw/ /ax/
/ax-h/ /axr/ /ay/ /eh/ /er/
/ey/ /ih/ /ix/ /iy/ /ow/ /oy/
/uh/ /uw/ /ux/

Vowels MacroClass

/l/ /r/ /w/ /y/ /
hh/ /hv/ /el/

SemiVowels MacroClass

/b/ /d/ /g/ /p/ /t/
/k/ /dx/ /q/ /bcl/
/dcl/ /gcl/ /pcl/
/tcl/ /kcl/

Stops MacroClass

/pau/ /epi/
/h#/

Others MacroClass

/m/ /n/ /ng/
/em/ /en/ /nx/

Nasals MacroClass

/s/ /sh/ /z/ /zh/
/f/ /th/ /v/ /dh/

Fricatives MacroClass

/ch/ /jh/

Affricates MacroClass